UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/471,072	12/21/1999	EDUARDO PELEGRI-LLOPART	SUN1P254/P41	6969

22434 7590 05/03/2004

BEYER WEAVER & THOMAS LLP
P.O. BOX 778
BERKELEY, CA 94704-0778

EXAMINER	
KISS, ERIC B	
ART UNIT	**PAPER NUMBER**
2122	20

DATE MAILED: 05/03/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Advisory Action	Application No. 09/471,072	Applicant(s) PELEGRI-LLOPART ET AL.	
	Examiner Eric B. Kiss	Art Unit 2122	

--The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

THE REPLY FILED 05 April 2004 FAILS TO PLACE THIS APPLICATION IN CONDITION FOR ALLOWANCE. Therefore, further action by the applicant is required to avoid abandonment of this application. A proper reply to a final rejection under 37 CFR 1.113 may only be either: (1) a timely filed amendment which places the application in condition for allowance; (2) a timely filed Notice of Appeal (with appeal fee); or (3) a timely filed Request for Continued Examination (RCE) in compliance with 37 CFR 1.114.

PERIOD FOR REPLY [check either a) or b)]

a) ☒ The period for reply expires 3 months from the mailing date of the final rejection.

b) ☐ The period for reply expires on: (1) the mailing date of this Advisory Action, or (2) the date set forth in the final rejection, whichever is later. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of the final rejection.
ONLY CHECK THIS BOX WHEN THE FIRST REPLY WAS FILED WITHIN TWO MONTHS OF THE FINAL REJECTION. See MPEP 706.07(f).

Extensions of time may be obtained under 37 CFR 1.136(a). The date on which the petition under 37 CFR 1.136(a) and the appropriate extension fee have been filed is the date for purposes of determining the period of extension and the corresponding amount of the fee. The appropriate extension fee under 37 CFR 1.17(a) is calculated from: (1) the expiration date of the shortened statutory period for reply originally set in the final Office action; or (2) as set forth in (b) above, if checked. Any reply received by the Office later than three months after the mailing date of the final rejection, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

1. ☐ A Notice of Appeal was filed on _____. Appellant's Brief must be filed within the period set forth in 37 CFR 1.192(a), or any extension thereof (37 CFR 1.191(d)), to avoid dismissal of the appeal.
2. ☐ The proposed amendment(s) will not be entered because:
   (a) ☐ they raise new issues that would require further consideration and/or search (see NOTE below);
   (b) ☐ they raise the issue of new matter (see Note below);
   (c) ☐ they are not deemed to place the application in better form for appeal by materially reducing or simplifying the issues for appeal; and/or
   (d) ☐ they present additional claims without canceling a corresponding number of finally rejected claims.
   NOTE: _____.
3. ☐ Applicant's reply has overcome the following rejection(s): _____.
4. ☐ Newly proposed or amended claim(s) _____ would be allowable if submitted in a separate, timely filed amendment canceling the non-allowable claim(s).
5. ☒ The a)☐ affidavit, b)☐ exhibit, or c)☒ request for reconsideration has been considered but does NOT place the application in condition for allowance because: *See Continuation Sheet*.
6. ☐ The affidavit or exhibit will NOT be considered because it is not directed SOLELY to issues which were newly raised by the Examiner in the final rejection.
7. ☒ For purposes of Appeal, the proposed amendment(s) a)☐ will not be entered or b)☒ will be entered and an explanation of how the new or amended claims would be rejected is provided below or appended.
   The status of the claim(s) is (or will be) as follows:
   Claim(s) allowed: _____.
   Claim(s) objected to: _____.
   Claim(s) rejected: *1,4,6-8,10 and 13-18*.
   Claim(s) withdrawn from consideration: _____.
8. ☒ The drawing correction filed on *05 April 2004* is a)☒ approved or b)☐ disapproved by the Examiner.
9. ☐ Note the attached Information Disclosure Statement(s)(PTO-1449) Paper No(s). _____.
10. ☐ Other: _____

WEI Y. ZHEN
PRIMARY PATENT EXAMINER

Continuation of 5. does NOT place the application in condition for allowance because:

Applicant's arguments are not persuasive.

The Examiner respectfully points out that one of ordinary skill in the computer art would recognize that any interpreting of code contained in a markup language document is equivalent to translating and executing said code. See, for example, the definitions of "interpret", "interpreted language", and "interpreter" on p. 285 of "Microsoft® Computer Dictionary," 2002, 5th ed., Microsoft Press, attached as part of Exhibit 1.

As stated in previous office actions, a *CFSET* object includes a mapping of scripting variables to values (see, for instance, the first example given on page 15 of the *CF Web* reference, which shows a mapping of the variable "FirstName" to the value "Jack"); the *GetBaseTagData()* method described in *CF Advanced* returns a list of available scripting variables and a variable type associated with each variable (as acknowledged by Applicant, the *GetBaseTagData()* method returns an object that contains all the variables, scopes, etc. of the nth ancestor with a given name. As the object returned contains available scripting variables and scope is a form of variable type, the *GetBaseTagData()* method meets the recited criteria of the TagExtraInfo object method); the GetBaseTagData function described in CF Advanced returns a list of available scripting variables and a variable type associated with each variable that is defined or modified by its associated action tag (GetBaseTagData returns an object that contains all the variables, scopes, etc. of the nth ancestor with a given name (see CF Advanced, p. 27 subsection titled "Ancestor data access"); as such, the variables defined by the associated ancestor tag are returned); the pageContext object of ColdFusion 4.0 includes a mapping of at least one scripting variable to a value that is or can be represented in the tag library (see, for example, CF Advanced, pp. 27-28 subsection titled "Example: Ancestor data access", which shows sample code snipped from a custom tag. The sample code shows multiple examples of the CFSET instruction used to create mappings involving previously established tag data).

Further, in response to Applicant's arguments regarding the claimed tag handler, the Examiner asserts that the cited sections of *CF Web* disclose instructions that, when executed, assign values to variables at runtime, thereby providing a runtime mapping of variables to values

In regard to the objection to the specification

Throughout prosecution, the Examiner has attempted to preserve the validity of the present Assignee's trademark JAVA by requesting that Applicant use the trademark in a manner consistent with MPEP §608.01(v) and with the present Assignee's document, "Sun Trademark and Logo Usage Requirements," (an Internet URL pointing to this document was provided in the Final Rejection mailed January 5, 2004 in the section titled Response to Amendment; additionally, the Examiner discussed the first requirement of

this document, namely, using trademarks as adjectives and not nouns, with Applicant's representative, Ramin Mahboubian (Reg. 44,890) during an Interview conducted on October 15, 2003, see the Interview Summary, mailed October 22, 2003).

Applicant has contested the Examiner's maintained objection to the specification based on improper usage of the trademark JAVA.

Specifically, Applicant has stated:

> It is respectfully submitted that Java as a programming language is known to those skilled in the art. As such, there is no need to accompany Java by "generic terminology." [see Amendment After Final, Paper No. 10, filed May 12, 2003, p. 5, paragraph 2].

> Furthermore, it is respectfully submitted the [sic] some programming languages or specific software products are well known to those skilled in the art by a trademark name (e.g., Java programming language). As such, it is NOT necessary to accompany trademark names such as the Java programming language with generic terminology. In fact, it may be inappropriate to use generic terminology if a programming language or product is primarily known to those skilled in the art by its trademark name (e.g., Java programming language). (Please see, for example, the Board of Appeals decision to the effect that a product on the market should be known to those skilled in the art or it be necessary to use the trade mark to identify the product; *Ex parte Frederick and Waterfall*, 75 USPQ 298 (Bd. Pat. App. & Int. 1947).) Nevertheless, solely in order to expedite prosecution, the trademark Java has been accompanied by generic terminology because of the Examiner's insistence. Clearly, the Applicant is NOT required to accompany every occurrence of the trademark name Java with generic terminology. Still furthermore, the Applicant is using its own trademark name. As such, the Applicant should not be required to use it in a way that the Examiner deems appropriate. [see Amendment After Final, Paper No. 19, filed April 5, 2004, p. 8, last paragraph, continuing onto p. 9; emphasis in original].

The Examiner acknowledges that MPEP §608.01(v) states, "Every effort should be made to prevent [the use of trademarks] in any manner which might adversely affect their validity as trademarks." The permissive language of this passage (recitation of "should" instead of, for example, "shall" or "must") suggests that Applicant may be permitted to use/misuse their own trademarks as they see fit (at their own peril), despite the objections made of record by the Examiner.

However, in order to fully address this issue, further clarification is required as to how Applicant intends the Office to interpret recitation of the term "Java" in the

instant specification. In the present case, Applicant apparently expressly contends (see Applicant's arguments from Paper Nos. 10 and 19, cited above) that "Java" itself is a well-known generic name for a programming language and its associated documentation, making any additional generic terminology unnecessary and redundant. If, as Applicant argues, JAVA has in fact become the generic name for a so-described programming language, then the objection to the specification, based on improper usage of trademarks, is deemed moot in view of the abandonment of the trademark JAVA (under 15 U.S.C. §1127) for describing a programming language and its associated documentation.

If, however, Applicant maintains that JAVA should be considered as a trademark rather than as a generic name used in trade (in contradiction to Applicant's above-cited arguments), then the trademark JAVA can only be descriptive of the source of the goods, and not the goods themselves. Accordingly, additional "generic terminology", e.g., "JAVA programming language" is necessary in order to properly describe (under 35 U.S.C. §112) the instant invention.

Attached **Exhibit 1**: "Microsoft® Computer Dictionary," 2002, 5th ed., Microsoft Press, pp. i, ii, 285.

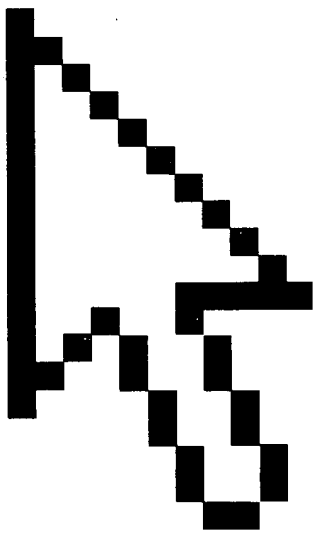
EXHIBIT 1

Microsoft

Microsoft®

Computer Dictionary

Fifth Edition



PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Copyright © 2002 by Microsoft Corporation

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Library of Congress Cataloging-in-Publication Data
Microsoft Computer Dictionary.--5th ed.
p. cm.
ISBN 0-7356-1495-4
1. Computers--Dictionaries. 2. Microcomputers--Dictionaries.

AQ76.5. M52267 2002
004'.03--dc21 200219714

Printed and bound in the United States of America.

2 3 4 5 6 7 8 9 QWT 7 6 5 4 3 2

Distributed in Canada by H.B. Fenn and Company Ltd.

A CIP catalogue record for this book is available from the British Library.

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office or contact Microsoft Press International directly at fax (425) 936-7329. Visit our Web site at www.microsoft.com/mspress. Send comments to *mspinput@microsoft.com*.

Active Desktop, Active Directory, ActiveMovie, ActiveStore, ActiveSync, ActiveX, Authenticode, BackOffice, BizTalk, ClearType, Direct3D, DirectAnimation, DirectDraw, DirectInput, DirectMusic, DirectPlay, DirectShow, DirectSound, DirectX, Entourage, FoxPro, FrontPage, Hotmail, IntelliEye, IntelliMouse, IntelliSense, JScript, MapPoint, Microsoft, Microsoft Press, Mobile Explorer, MS-DOS, MSN, Music Central, NetMeeting, Outlook, PhotoDraw, PowerPoint, SharePoint, UltimateTV, Visio, Visual Basic, Visual C++, Visual FoxPro, Visual InterDev, Visual J++, Visual SourceSafe, Visual Studio, Win32, Win32s, Windows, Windows Media, Windows NT, Xbox are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other product and company names mentioned herein may be the trademarks of their respective owners.

The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

Acquisitions Editor: Alex Blanton
Project Editor: Sandra Haynes

Body Part No. X08-41929

InterNIC *n.* Short for NSFnet (Internet) Network Information Center. The organization that has traditionally registered domain names and IP addresses as well as distributed information about the Internet. InterNIC was formed in 1993 as a consortium involving the U.S. National Science Foundation, AT&T, General Atomics, and Network Solutions, Inc. (Herndon, Va.). The latter partner administers InterNIC Registration Services, which assigns Internet names and addresses.

interoperability *n.* Referring to components of computer systems that are able to function in different environments. For example, Microsoft's NT operating system is interoperable on Intel, DEC Alpha, and other CPUs. Another example is the SCSI standard for disk drives and other peripheral devices that allows them to interoperate with different operating systems. With software, interoperability occurs when programs are able to share data and resources. Microsoft Word, for example, is able to read files created by Microsoft Excel.

interpolate *vb.* To estimate intermediate values between two known values in a sequence.

interpret *vb.* **1.** To translate a statement or instruction into executable form and then execute it. **2.** To execute a program by translating one statement at a time into executable form and executing it before translating the next statement, rather than by translating the program completely into executable code (compiling it) before executing it separately. *See also* interpreter. *Compare* compile.

interpreted language *n.* A language in which programs are translated into executable form and executed one statement at a time rather than being translated completely (compiled) before execution. Basic, LISP, and APL are generally interpreted languages, although Basic can also be compiled. *See also* compiler. *Compare* compiled language.

interpreter *n.* A program that translates and then executes each statement in a program written in an interpreted language. *See also* compiler, interpreted language, language processor.

interprocess communication *n.* The ability of one task or process to communicate with another in a multitasking operating system. Common methods include pipes, semaphores, shared memory, queues, signals, and mailboxes. *Acronym:* IPC.

inter-record gap *n.* An unused space between data blocks stored on a disk or tape. Because the speed of disks and tapes fluctuates slightly during operation of the drives, a new data block may not occupy the exact space occupied by the old block it overwrites. The inter-record gap prevents the new block from overwriting part of adjacent blocks in such a case. *Acronym:* IRG. *Also called:* gap, interblock gap.

interrogate *vb.* To query with the expectation of an immediate response. For example, a computer may interrogate an attached terminal to determine the terminal's status (readiness to transmit or receive).

interrupt *n.* A signal from a device to a computer's processor requesting attention from the processor. When the processor receives an interrupt, it suspends its current operations, saves the status of its work, and transfers control to a special routine known as an interrupt handler, which contains the instructions for dealing with the particular situation that caused the interrupt. Interrupts can be generated by various hardware devices to request service or report problems, or by the processor itself in response to program errors or requests for operating-system services. Interrupts are the processor's way of communicating with the other elements that make up a computer system. A hierarchy of interrupt priorities determines which interrupt request will be handled first if more than one request is made. A program can temporarily disable some interrupts if it needs the full attention of the processor to complete a particular task. *See also* exception, external interrupt, hardware interrupt, internal interrupt, software interrupt.

interrupt-driven processing *n.* Processing that takes place only when requested by means of an interrupt. After the required task has been completed, the CPU is free to perform other tasks until the next interrupt occurs. Interrupt-driven processing is usually employed for responding to events such as a key pressed by the user or a floppy disk drive that has become ready to transfer data. *See also* interrupt. *Compare* autopolling.

interrupt handler *n.* A special routine that is executed when a specific interrupt occurs. Interrupts from different causes have different handlers to carry out the corresponding tasks, such as updating the system clock or reading the keyboard. A table stored in low memory contains pointers, sometimes called vectors, that direct the processor to the various interrupt handlers. Programmers can create interrupt handlers to replace or supplement existing handlers,